

**In The Claims**

✓ / / /  
Please cancel claims 4, 5, 9 and 10 without prejudice to or disclaimer of the subject matter contained therein.

Please amend the claims as follows:

Sub  
DI  
1. (Amended) In a method for implementing an event transfer system of a real time operating system kernel under a multi-tasking environment in which a priority-based preemptive scheduling is adapted, a method for implementing an event transfer system of a real time operating system kernel, comprising:

calling, by each of a plurality of tasks, a kernel system function of receiving an event with respect to one event under the multi-tasking environment; and

blocking said each of the tasks and inserting said each of the tasks into a waiting-list of the event in priority order when no event is provided to the tasks,

wherein in the case that the event transfer occurs, the task having the highest priority in the waiting-list obtains the event, is woken up and is resumed with execution.

2. (Amended) The method of claim 1, wherein said waiting-list of the event is managed based on the priority order so that the task having the

b10  
highest priority is arranged at the most leading portion (head) of the waiting-list.

3. (Amended) The method of claim 1, further comprising:

checking whether there is an event value already sent, when the kernel system function of receiving the event starts.

---

b11  
6. (Amended) The method of claim 1, further comprising:

checking a validity of the event ID for thereby generating an error code in the case of validity when the kernel system function of receiving the event starts; and

returning the routine from the kernel system function.

7. (Amended) The method of claim 13, wherein when the current task is queued into the waiting-list, a time out option is additionally set if it exists.

8. (Amended) The method of claim 1, wherein when transferring the event, the method further comprises:

checking whether any waiting task exists in the waiting-list of the event.

---

b12  
11. (Amended) The method of claim 15, wherein said head task in the waiting-list which receives the event value is adjusted to a ready state and is

inserted into a ready list, and an additional process routine by the sort of the event is executed.

b12 12. (Amended) The method of claim 3, wherein as a result of the check whether the event value exists, when the event value exists, the event value is obtained in the event control block buffer, and the task routine is executed by the sort of the event.

---

14. (Amended) The method of claim 3, wherein when the kernel system function of receiving the event starts, the method further comprises:

b13 checking a validity of an event ID for thereby generating an error code in the case of invalidity; and  
returning the routine from the kernel system function.

15. (Amended) The method of claim 8, wherein as a result of the check that whether the task exists in the waiting-list, when the waiting task does not exist, an event value is stored in an event buffer of an event control block.

---

Please add the following new claims:

b14 --17. (New) A method for implementing an event transfer system of a real time operating system kernel, the method comprising:

calling, by each of a plurality of tasks, a kernel system function of receiving an event with respect to one event in a multi-tasking environment; and

blocking execution of said each of the tasks and inserting said each of the tasks into a waiting-list of the event according to a priority order when no event is provided to the tasks.

b14  
18. (New) The method of claim 17, further comprising:  
accessing the task having the highest priority from the waiting-list of event when the event is provided; and  
executing the accessed task.

19. (New) The method of claim 17, further comprising:  
checking whether an event value exist when the kernel system function of receiving the event starts.

20. (New) The method of claim 19, wherein as a result of said checking, the current task is blocked and queued into the waiting-list of the event according to the priority order if the event value does not exist.

21. (New) The method of claim 19, further comprising:

obtaining the event value from an event control block storage and executing the task routine when said checking indicates that the event value exists.

22. (New) The method of claim 17, wherein when the kernel system function of receiving the event starts, the method further comprises:

b14 checking a validity of an event ID to thereby generate an error code in the case of invalidity; and

returning the routine from the kernel system function.

23. (New) The method of claim 17, further comprising:

checking whether any waiting task exists in the waiting-list of the event when transferring the event; and

transferring an event value to the head task of the waiting-list when said checking indicates that the waiting task exists in the waiting-list.

24. (New) The method of claim 23, wherein said head task in the waiting-list which receives the event value is adjusted to a ready state and is inserted into a ready list, and an additional process routine by the sort of the event is executed.--

---